Supplementary Material for AvatarPose: Avatar-guided 3D Pose Estimation of Close Human Interaction from Sparse Multi-view Videos

Feichi Lu[1*], Zijian Dong[1,2*], Jie Song[1†], and Otmar Hilliges[1]

[1] Department of Computer Science, ETH Zürich, Switzerland,
[2] Max Planck Institute for Intelligent Systems, Germany

This **Supplementary Material** document provides additional details and experimental results as mentioned in the main paper. In Section 1, we provide further implementation details of our proposed method, including additional training objectives, sampling strategy, and more training strategy and parameter setting details. Section 2 explains details of the implementation of baseline methods and evaluation metrics. In Section 3, we provide additional qualitative results and comparisons (Section 3.1 and Section 3.2). We also demonstrate the robust performance of our method with 4 views (Section 3.3). Section 3.5 shows the results of additional ablation studies for the alternating optimization. Further experiments for the avatar training and rendering are presented in Section 3.6 and Section 3.7. Finally, we discuss our limitations and possible future works in Section 4.

In addition, please see the **Supplementary Video**, which better illustrates our method and results for the task of estimating 3D poses of closely interacting people.

1 Implementation

1.1 Technical Details

Avatar Training Objectives During the multi-avatar prior learning, we follow [7] to add the hard surface regularization term :

$$\mathcal{L}_{hard} = -\frac{1}{\mid \mathcal{R} \mid} \sum_{\mathbf{r} \in \mathcal{R}} \log(e^{|\alpha(\mathbf{r})|} + e^{|\alpha(\mathbf{r})-1|}) + const, \tag{1}$$

where $\alpha(\mathbf{r})$ is the ray opacity calculated from $\alpha(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i \Pi_{j<i}(1-\alpha_j), \alpha_i = 1 - \exp(-\sigma_i \delta_i)$. $\mathbf{r} \in \mathcal{R}$ where $\mathcal{R}$ is the set of sampled rays. σ_i is the density of the sampled point $\mathbf{x}_i$ and δ_i is the distance between samples along the ray $\mathbf{r}$.

[*] Equal contribution
[†] Now at HKUST(GZ)&HKUST

We also enforce a density regularization similar to the one proposed by [7].

$$\begin{aligned} \mathcal{L}_{density} &= -\frac{1}{\mid \mathcal{W} \mid} \sum_{w_i \in \mathcal{W}} \log(e^{-w_i} + e^{w_i - 1}) \\ w_i &= \alpha_i \Pi_{j<i}(1-\alpha_j) \quad \mathcal{W} = \{w_i\} \end{aligned} \tag{2}$$

Here, w_i is the weight of density for each sampled point $\mathbf{x}_i$ along the rays, and $\mathcal{W}$ is the set of all weights w_i for all samples $\mathbf{x}_i$ along all rays $\mathbf{r} \in \mathcal{R}$.

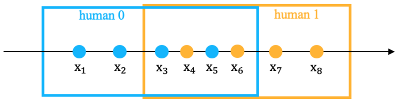


Fig. 1: Layered Rendering. For a ray $\mathbf{r}$ marching through the 3D bounding box of human instances, we uniformly sample points along the ray for each instance. If the sampling regions of different instances intersect with each other, we sort the sampled points by the depth values and compute the color of the ray following the sorted order of points via numerical integration [10].

Layered Volume Rendering Fig. 1 illustrates the detailed sampling, sorting, and rendering process for a ray in layered rendering. We first sample points in the bounding box of each human, and then calculate their corresponding color and density from each avatar model respectively. For each human, we sample 256 points along the ray. Following [7], a sampled point has zero density when it falls into an empty cell in the occupancy grid. All sampled points are sorted and rendered together to get the final color of the ray.

Alternating Optimization In the stage of avatar optimization, we first optimize the appearance of avatars using the whole sequence for 10 epochs. To alleviate the negative effect of inaccurate poses on the avatar appearance, we select frames where estimated poses tend to be correct and leverage them to refine the avatar for another 20 epochs. We empirically observe that when the people are in close interaction, the estimated poses are prone to have larger errors. Motivated by this observation, we calculate the average distance of people from the estimated initial pose and then select frames where the personal distance is relatively large to refine avatars. In the stage of pose optimization, we fix the parameters of the avatar network and only update the SMPL parameters $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}^{(l)}\}_{l \in [1,L]}$, where L is the number of humans in the scene. We first train the whole SMPL model for 10 epochs, and then train poses for arms and hands using the semantic-guided sampler for another 10 epochs.

We alternatively train the avatar stage and pose stage for 3 rounds, and then perform a final stage of avatar training to get our final result.

Ray Sampling Strategy During pose optimization, uniform sampling of rays is inefficient, since most of the loss changes occur close to the edge of the human mask. Inspired by this insight, we leverage SAM-Track [2] to get the mask for all humans and sample more pixels close to the edge of the mask during optimization. Additionally, it is challenging to optimize the body parts such as arms or hands when they are in close contact. To improve the pose, we leverage body part segmentation in the SAM-Track mask and sample more points in this region. We use an edge sampler to sample more pixels close to the edge of the SAM-Track human mask, the sample range is set to be $K = 48$ pixels in and out of the edge of the mask so that enough positive and negative points would be included.

SMPL initialization Since InstantAvatar relies on SMPL pose parameters to become animatable, we need to estimate a proper initialization for the SMPL parameters. We first estimate 19 joints from the multi-view videos using 4DAssociation [22]. Then we fit SMPL parameters to the joints using the SMPL estimation code released by easymocap [14].

1.2 Training Details

We train our networks using the Adam optimizer [9], with different learning rates for appearance ($l = 1e^{-3}$), pose ($l = 1e^{-4}$), rotation and translation ($l = 1e^{-5}$). Our model can be trained on a single NVIDIA RTX 3090 GPU.

We also use different weights for different loss terms. In the avatar optimization stage, we apply weight 1.0 for RGB loss, 0.1 for silhouette loss, 0.1 for hard surface regularization, 0.001 for instance layer regularization, and 0.1 for density regularization. In the pose optimization stage, we use weight 10.0 for RGB loss, 0.05 for silhouette loss, 0.001 for pose regularization, and 0.001 for penetration regularization.

2 Evaluation Details

2.1 Data

We evaluate our method on Hi4D [21], CHI3D [6], MultiHuman Real-Cap [23], Shelf [1], and Panoptic [8] datasets.

Hi4D The released dataset has 8 calibrated cameras with corresponding captured videos of 2 people in close interaction, for example, hugging, dancing, and doing sports. It includes registrations of SMPL body parameters which can be considered as ground truth and from which we can infer the ground truth 3D joint locations. It also contains ground truth instance masks, but here we only use the estimated combined human mask from SAM-Track [2]. We use all the sequences in Hi4D for evaluation.

CHI3D This dataset has 4 calibrated cameras with corresponding videos capturing close interactions between 2 people. It captures interactions like grabbing, handshaking, and hitting. It has SMPLX ground-truth registration with one person in the scene and the pseudo-ground-truth SMPLX for the other person. We sample 72 sequences uniformly distributed in each group of interactions in CHI3D.

MultiHuman Real-Cap This dataset contains a sequence with 3 people interacting with each other, with some body contacts. It has 6 calibrated cameras capturing videos. It has no ground-truth SMPL registration.

Shelf Shelf is a popular dataset used for evaluating multi-view multi-person pose estimation. However, it seldom involves close contact among the 4 people captured in the scene. The videos are captured with 5 calibrated cameras and for some frames are annotated with ground-truth 3D skeleton positions.

Panoptic The Panoptic dataset contains different sequences with different numbers of people. It does not involve close human-body contact. There are 5 calibrated cameras and ground-truth 3D skeleton annotations.

2.2 Baseline

For pure association-based methods, we choose 4DAssociation [22] and MVPose* [5]. Based on MVPose* [5], MVPose [4] added temporal tracking and SMPL prior to the pipeline, which is regarded as SMPL-guided category. Since MVpose [4,5] mainly uses 25 joints, we use a subset mapping to select the 19 joints used in our method. For the regression-based methods Faster VoxelPose [20], MvP [17], Graph [18] and Tempo [3], we fine-tune the provided backbone with a small subset of sequences in Hi4D and CHI3D and test on the rest of the sequences. Since they utilize a system of 15 joints introduced by the Panoptic Dataset [8], we first fit a SMPL model to the estimated 15 joints, then extract the 19 joints in our method from the fitted SMPL model, so that all the methods are evaluated with the same 19 joints.

2.3 Evaluation Metric

The Mean Per Joint Position Error (MPJPE) is used to calculate the average distance between the ground truth and estimated joints. Based on MPJPE, Percentage of Correct Parts (PCP3D) finds the closest pose estimation for each ground truth pose and computes the percentage of correct parts. Since PCP3D does not penalize false positive pose estimation, we add the Average Precision (AP_K) metric from [16]. When MPJPE is smaller than K millimeters, we consider the corresponding pose as accurate.

Method	Actor1	Actor2	Actor3	Avg
Graph [18]	99.3	96.5	97.3	**97.7**
MvP [17]	99.3	95.1	**97.8**	97.4
Faster VoxelPose [20]	**99.4**	96.0	97.5	97.6
MVPose [4]	98.8	94.1	**97.8**	96.9
4DAssociation [22]	99.0	96.2	97.6	97.6
Ours	98.4	**96.9**	**97.8**	**97.7**

Table 1: Quantitative Comparison with SotA on the Shelf [21] Dataset. We compare our method with Graph [19], MvP [17], Faster VoxelPose [20], MVPose* [5], MVPose [4] and 4DAssociation [22]. We use the metric percentage of correct parts (PCP3D) used by all the previous methods.

3 Additional Results

3.1 Comparison to Baselines

In Figure 3 and Figure 4, we show additional qualitative comparison with other baselines. While other methods encounter the problems of abnormal 3D joint detection, interpenetration, and missing and redundant detections, our method achieves more robust and accurate performance in all of the challenging sequences on Hi4D.

In Table 1, we compare our method with the SotA methods on the Shelf dataset. We show that our method outperforms most previous methods and achieves the highest average PCP3D metric. We also compare our method quantitatively and qualitatively with Tempo [3] in Table 2, Figure 3 and Figure 4. Tempo is a learning-based method that tends to overfit the pose distribution in the training subset. This can cause misaligned joint estimations or confusion among keypoints. Tempo also has the problem of missing people when people are in close interaction.

Dataset	**Method**	**MPJPE**(mm) ↓	**PCP**(%) ↑	**AP**$_{50}$ ↑	**AP**$_{100}$ ↑	**Recall**(%) ↑
Hi4D	Tempo [3]	52.70	83.24	57.68	80.55	89.83
Hi4D	Ours	**32.98**	**99.79**	**93.20**	**99.79**	**99.85**
CHI3D	Tempo [3]	52.39	87.61	62.29	87.84	96.08
CHI3D	Ours	**32.10**	**96.90**	**91.48**	**97.33**	**98.78**

Table 2: Quantitative Comparison with SotA on Hi4D [21] and CHI3D [6]. We compare our method with Tempo [3]. We report MPJPE, PCP, AP$_K$, and Recall metric for all methods.

3.2 Additional Qualitative Results

Figure 5 and Figure 6 show additional qualitative results of our method on different motion sequences in Hi4D with close body interaction. Figure 7 shows additional qualitative results of our method on different sequences in CHI3D. Figure 8 shows additional qualitative results of our method on Shelf. Figure 9 shows additional qualitative results on MultiHuman Real-Cap and Panoptic datasets.

3.3 Robustness to Number of Views

In Table 3, we report the comparison between our method and one SotA method 4DAssociation [22] under different numbers of views. Our method achieves robust performance when the number of cameras becomes very small (4 views). Notably, our method with 4 views even outperforms the SotA method with 8 views.

Method	**MPJPE**(mm) ↓
4DAssociation (4views)	49.45
4DAssociation (8views)	40.44
Ours (4 views)	**34.26**
Ours (8 views)	**29.37**

Table 3: Robustness to Number of Views. Ablations to evaluate the performance of our method on Hi4D with 4 views and 8 views. For comparison, we also report results of 4DAssociation [22].

3.4 Ablation Study on Penetration Loss

Table 4 shows a quantitative ablation study to prove the effectiveness of the penetration loss. Training with the penetration loss decreases the average MPJPE loss by 0.2mm. Despite the minor numerical improvement, this loss greatly helps avoid collisions.

Method	**MPJPE**(mm) ↓	**PCP**(%) ↑	**AP**$_{50}$ ↑	**AP**$_{100}$ ↑
Ours (w/o penetration loss)	29.91	96.98	87.72	**97.13**
Ours	**29.71**	**97.05**	**87.87**	**97.13**

Table 4: Ablation Study on Penetration Loss

3.5 Alternating Optimization Combining Joint Optimization

We also try to combine alternating optimization with joint optimization, with one alternating optimization stage followed by a joint optimization stage. As is shown in Table 5, with alternating optimization combined with joint optimization, the average joint position error becomes smaller and the percentage of correct parts is higher. However, according to the precision metrics, the AP_{50} and AP_{100} get lower, which means there are more outliers after the joint optimization. When the alternating optimization does not fully correct the wrong poses, joint optimization can have a negative effect on pose optimization.

Method	**MPJPE**(mm) ↓	**PCP**(%) ↑	$\textbf{AP}_{50}$ ↑	$\textbf{AP}_{100}$ ↑	**Recall**(%) ↑
Ours (Alternating Only)	31.77	91.62	**92.42**	**98.57**	98.57
Ours (Alternating then Joint)	**31.06**	**92.40**	92.25	96.43	98.57

Table 5: Ablation Study: Alternating Optimization, Alternating then joint optimization

3.6 Significance of Layered Rendering

Although SAM-Track [2] has reached state-of-the-art performance to segment and track instances in videos, it still fails to segment different human instances in close interaction. From Fig. 2, we can see that SAM-Track suffers from occlusions and close interactions and cannot segment small body parts such as hands and arms correctly. The problem of instance segmentation makes it hard to reconstruct avatars separately using single-person avatar models, e.g. InstantAvatar [7]. However, the combined mask for all humans in close interactions is observed to be accurate. Thus, we choose layered rendering to train multiple avatars together.

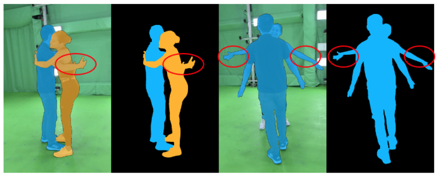


Fig. 2: SAM-Track Human Mask. SAM-Track tends to fail in human individual segmentation when people are in close interaction. This leads to artifacts when we train single avatars separately using individual instance masks. Our method leverages layered rendering, which uses the combined mask of all humans in the scene, which is always much more accurate than the individual masks.

3.7 Training and Rendering Speed for Avatars

Our multi-avatar prior training and rendering process achieves relatively fast speed due to the fast nature of Instant NGP [11]. The fast training and rendering speed for avatars enables the alternating optimization between avatars and poses. According to Table 6, our method is 20× faster in training and 8× faster in rendering than the state-of-the-art multi-human reconstruction method [14]. Both methods are experimented on a single NVIDIA RTX 3090 GPU.

Method	**Training** (h) ↓	**Rendering** (s) ↓
Shuai et al. [14]	9.7	3.93
Ours	**0.5**	**0.47**

Table 6: Avatar Training Speed. Compared with [14], our model achieves faster training and rendering speed for avatars, enabling alternating optimization between avatars and poses. (Rendering time here means average rendering time for all avatars in one frame.)

4 Limitations and Future Work

Although our method significantly outperforms previous methods, it still has several limitations to improve in the future. We do not model hands in our avatar model, which leads to inaccurate registration of 3D hand poses. It will be a promising direction in the future to integrate hand models [12,13] into our personalized avatar. The optimization pipeline of our current method is not very fast and it can also be accelerated by combining more efficient representation [24] or leveraging more powerful optimization tool [15]. Currently, we only estimate the 3D shape and pose of closely interacting people. Extending our idea into human-object interaction can be a promising direction to explore in the future. We also believe it will be interesting future work to adapt our pipeline to estimate 3D poses of closely interacting people from in-the-wild and monocular videos and images.

References

1. Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., Ilic, S.: 3d pictorial structures for multiple human pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1669–1676 (2014)
2. Cheng, Y., Li, L., Xu, Y., Li, X., Yang, Z., Wang, W., Yang, Y.: Segment and track anything. arXiv preprint arXiv:2305.06558 (2023)
3. Choudhury, R., Kitani, K.M., Jeni, L.A.: Tempo: Efficient multi-view pose estimation, tracking, and forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14750–14760 (2023)

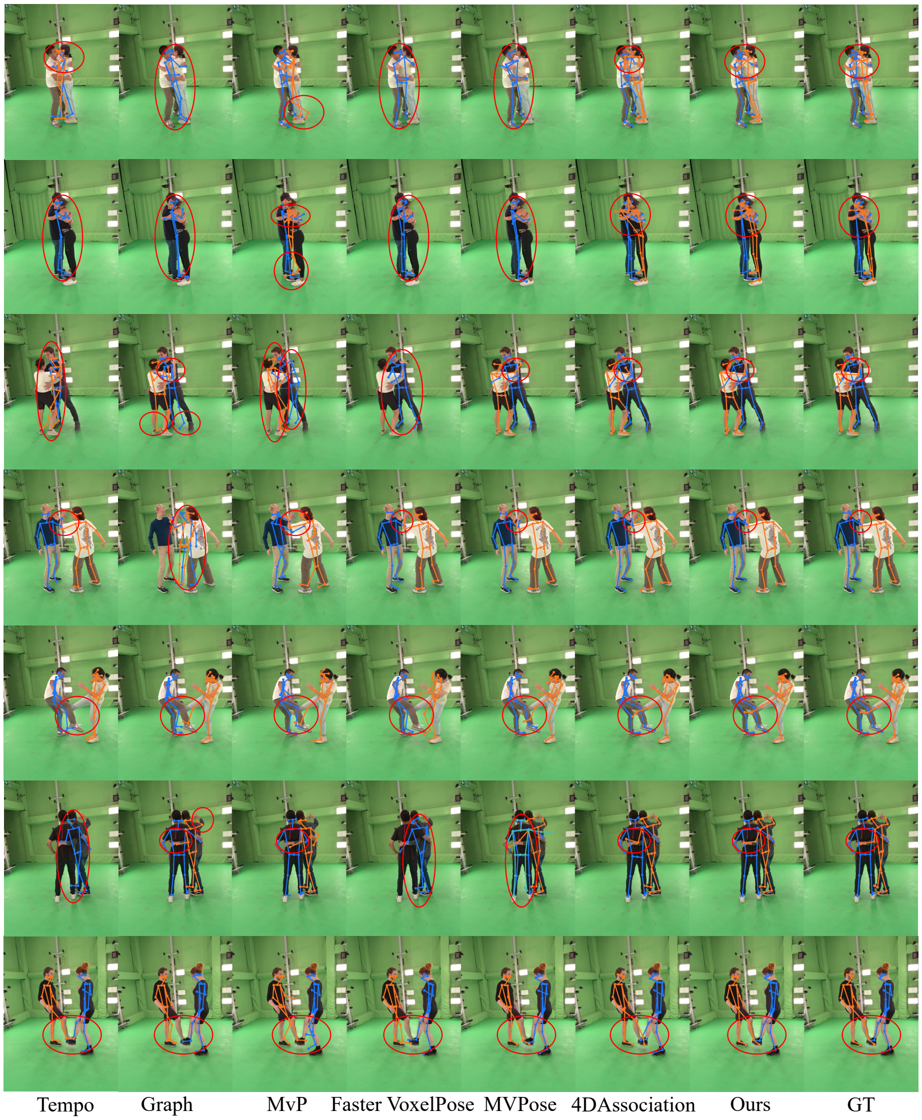


Fig. 3: Additional Qualitative Comparison Results. We present six examples from Hi4D dataset, comparing our method with Tempo [3], Graph [18], MvP [17], Faster VoxelPose [20], MVPose [4], and 4DAssociation [22], all using 8 views. In our illustrations, we use red circles to point out issues in other methods and to showcase the corresponding results from our approach.

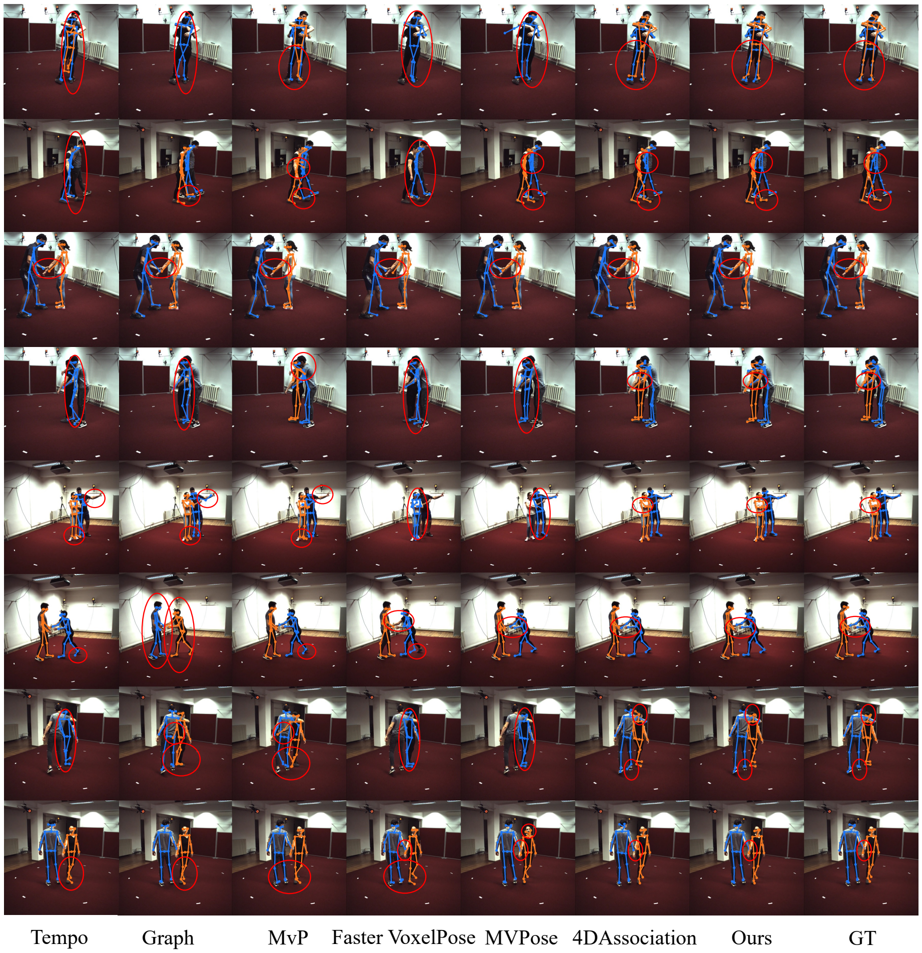


Fig. 4: Additional Qualitative Comparison Results. We present six examples from CHI3D dataset, comparing our method with Tempo [3], Graph [18], MvP [17], Faster VoxelPose [20], MVPose [4], and 4DAssociation [22], all using 4 views. In our illustrations, we use red circles to point out issues in other methods and to showcase the corresponding results from our approach.

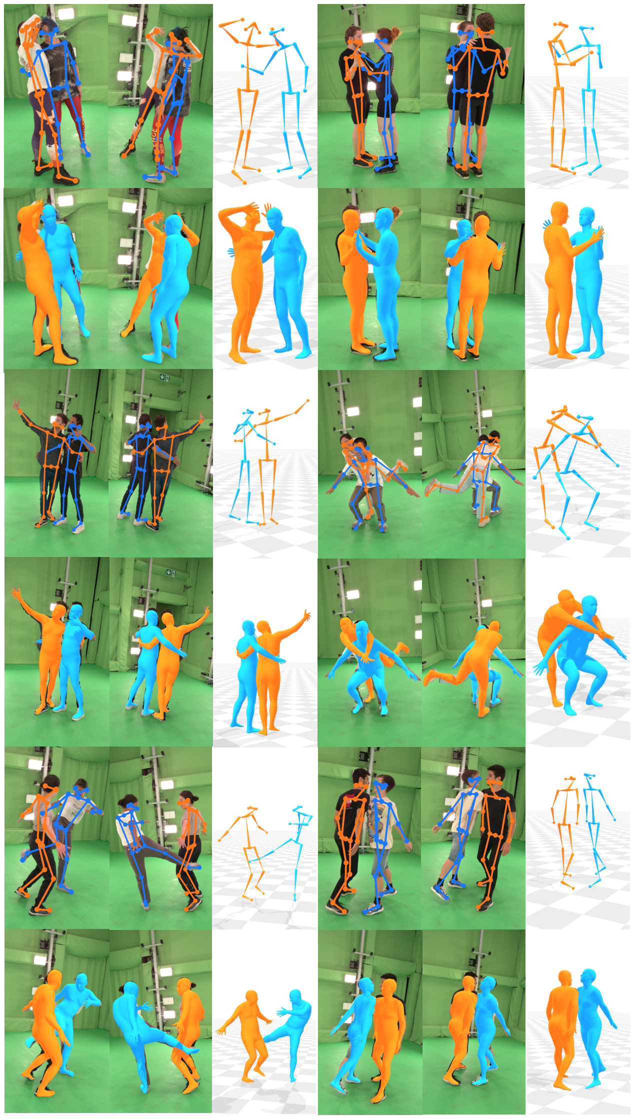


Fig. 5: Additional Qualitative Results. We show additional qualitative results of our method on the Hi4D dataset [21]. The left and middle columns show the 2D projections of the estimated 3D skeletons and SMPL body meshes on two views. The right column demonstrates skeletons and SMPL bodies in 3D scenes.

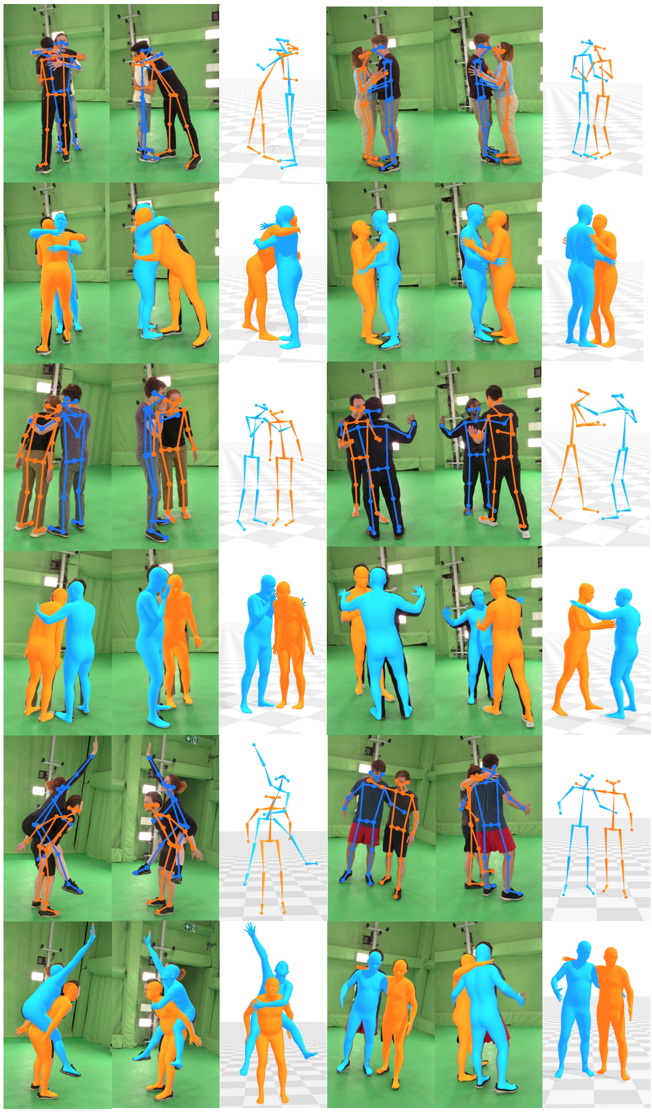


Fig. 6: Additional Qualitative Results. We show additional qualitative results of our method on the Hi4D dataset [21]. The left and middle columns show the 2D projections of the estimated 3D skeletons and SMPL body meshes on two views. The right column demonstrates skeletons and SMPL bodies in 3D scenes.

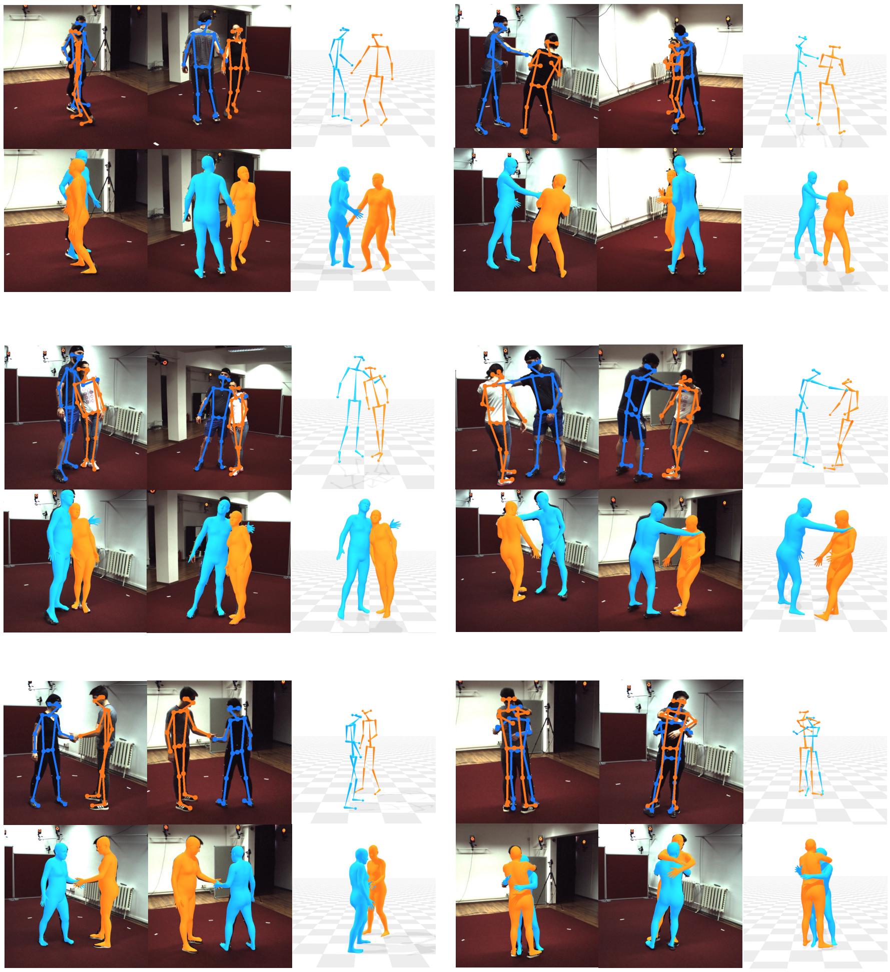


Fig. 7: Additional Qualitative Results. We show additional qualitative results of our method on the CHI3D dataset [6]. The left and middle columns show the 2D projections of the estimated 3D skeletons and SMPL body meshes on two views. The right column demonstrates skeletons and SMPL bodies in 3D scenes.

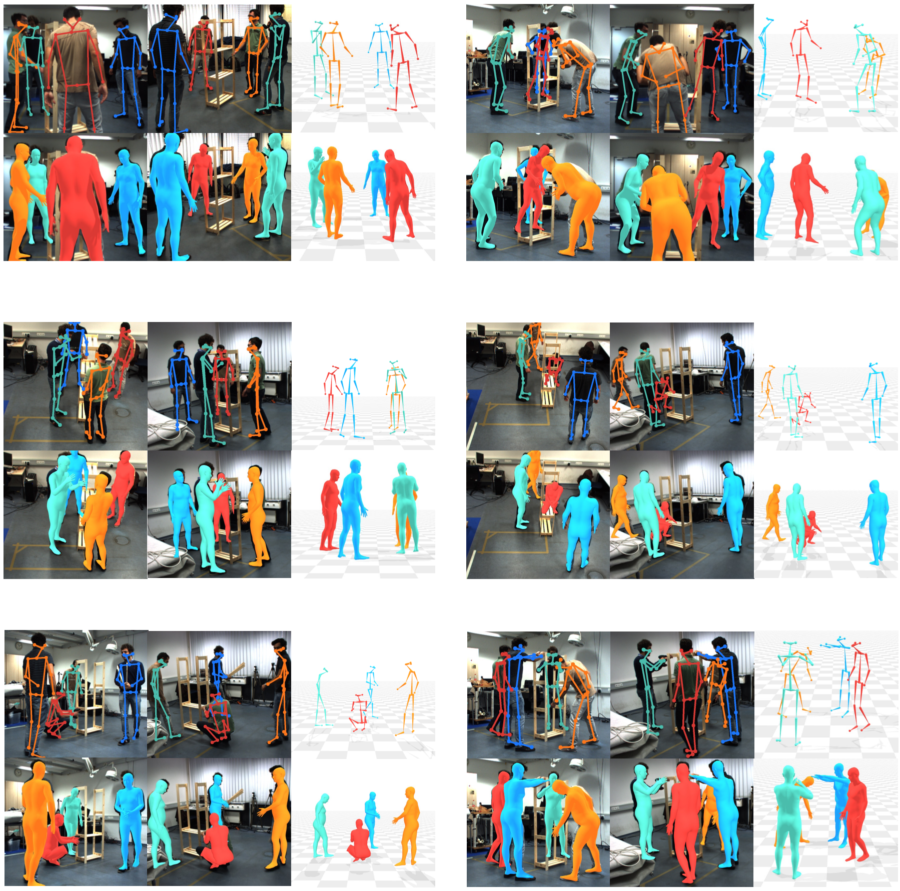
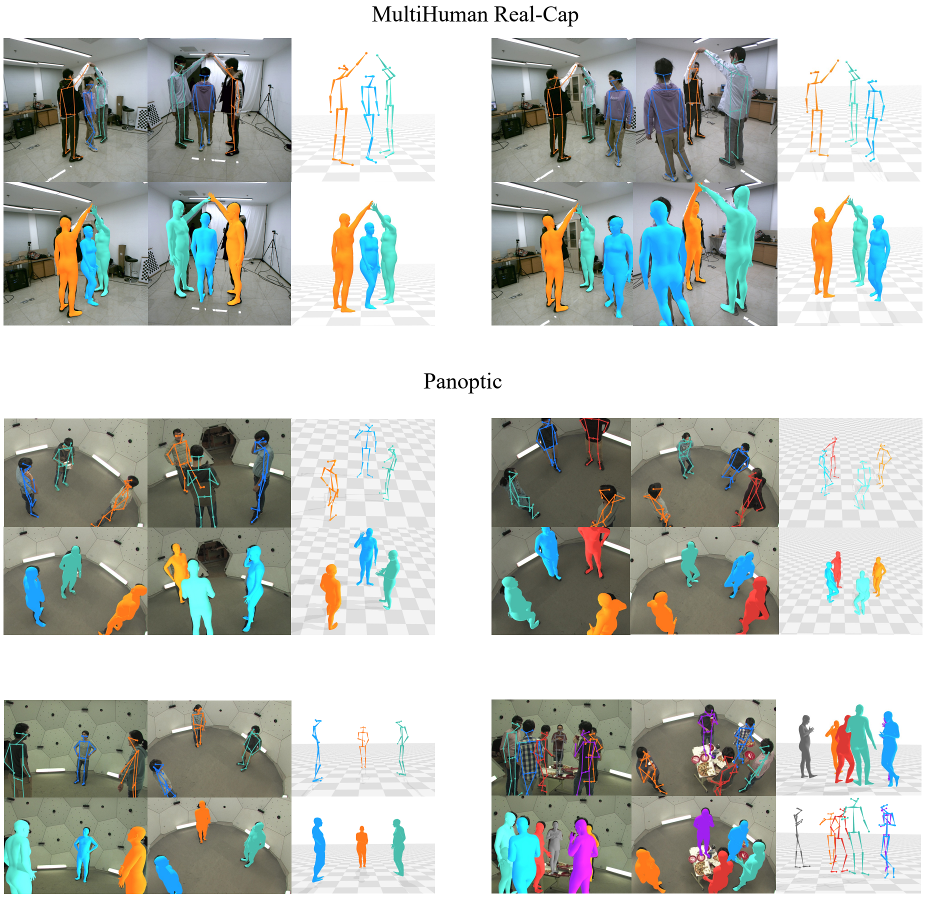


Fig. 8: Additional Qualitative Results. We show additional qualitative results of our method on the Shelf dataset [1]. The left and middle columns show the 2D projections of the estimated 3D skeletons and SMPL body meshes on two views. The right column demonstrates skeletons and SMPL bodies in 3D scenes.



MultiHuman Real-Cap

Panoptic

Fig. 9: Additional Qualitative Results. We show additional qualitative results of our method on the MultiHuman Real-Cap dataset [23] and Panoptic dataset [8]. The left and middle columns show the 2D projections of the estimated 3D skeletons and SMPL body meshes on two views. The right column demonstrates skeletons and SMPL bodies in 3D scenes.

4. Dong, J., Fang, Q., Jiang, W., Yang, Y., Bao, H., Zhou, X.: Fast and robust multi-person 3d pose estimation and tracking from multiple views. In: T-PAMI (2021)
5. Dong, J., Jiang, W., Huang, Q., Bao, H., Zhou, X.: Fast and robust multi-person 3d pose estimation from multiple views. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7792–7801 (2019)
6. Fieraru, M., Zanfir, M., Oneata, E., Popa, A.I., Olaru, V., Sminchisescu, C.: Three-dimensional reconstruction of human interactions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7214–7223 (2020)
7. Jiang, T., Chen, X., Song, J., Hilliges, O.: Instantavatar: Learning avatars from monocular video in 60 seconds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16922–16932 (2023)
8. Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social motion capture. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3334–3342 (2015)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
11. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
12. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3d hands, face, and body from a single image. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10975–10985 (2019)
13. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. arXiv preprint arXiv:2201.02610 (2022)
14. Shuai, Q., Geng, C., Fang, Q., Peng, S., Shen, W., Zhou, X., Bao, H.: Novel view synthesis of human interactions from sparse multi-view videos. In: ACM SIGGRAPH 2022 Conference Proceedings. SIGGRAPH '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3528233.3530704, https://doi.org/10.1145/3528233.3530704
15. Song, J., Chen, X., Hilliges, O.: Human body model fitting by learned gradient descent. In: European Conference on Computer Vision. pp. 744–760. Springer (2020)
16. Tu, H., Wang, C., Zeng, W.: Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. pp. 197–212. Springer (2020)
17. Wang, T., Zhang, J., Cai, Y., Yan, S., Feng, J.: Direct multi-view multi-person 3d pose estimation. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 13153–13164. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper_files/paper/2021/file/6da9003b743b65f4c0ccd295cc484e57-Paper.pdf
18. Wu, S., Jin, S., Liu, W., Bai, L., Qian, C., Liu, D., Ouyang, W.: Graph-based 3d multi-person pose estimation using multi-view images. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11148–11157 (2021)
19. Wu, S., Jin, S., Liu, W., Bai, L., Qian, C., Liu, D., Ouyang, W.: Graph-based 3d multi-person pose estimation using multi-view images. In: ICCV (2021)

20. Ye, H., Zhu, W., Wang, C., Wu, R., Wang, Y.: Faster voxelpose: Real-time 3d human pose estimation by orthographic projection. In: European Conference on Computer Vision. pp. 142–159. Springer (2022)
21. Yin, Y., Guo, C., Kaufmann, M., Zarate, J.J., Song, J., Hilliges, O.: Hi4d: 4d instance segmentation of close human interaction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17016–17027 (2023)
22. Zhang, Y., An, L., Yu, T., Li, X., Li, K., Liu, Y.: 4d association graph for realtime multi-person motion capture using multiple video cameras. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1324–1333 (2020)
23. Zheng, Y., Shao, R., Zhang, Y., Yu, T., Zheng, Z., Dai, Q., Liu, Y.: Deepmulticap: Performance capture of multiple characters using sparse multiview cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6239–6249 (2021)
24. Zheng, Z., Zhao, X., Zhang, H., Liu, B., Liu, Y.: Avatarrex: Real-time expressive full-body avatars. arXiv preprint arXiv:2305.04789 (2023)